



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

PETTERI LEHTINEN
DATA MIGRATION TO A NEXT GENERATION INTEGRATION
MIDDLEWARE
Master's Thesis

Examiner: Professor Tommi Mikkonen
Examine and Topic approved in the
faculty of Computing and Electrical
Engineering council meeting on 6th
April 2016

TIIVISTELMÄ

PETTERI LEHTINEN: Datamigraatio seuraavan sukupolven integraatio-ohjelmistoon

Tampereen Teknillinen Yliopisto

Diplomityö, 46 sivua

Toukokuu 2016

Tietotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Ohjelmistotuotanto

Tarkastaja: professori Tommi Mikkonen

Avainsanat: yritysten tietojärjestelmät, integraatio-ohjelmisto, datamigraatio

Yritykset käyttävät tietojärjestelmiä osana kriittisiä liiketoimintaprosessejaan. Näiden järjestelmien tehokas integrointi parantaa prosessien tehokkuutta ja siten yrityksen kilpailukykyä. Liiketoimintaprosessit vaativat paitsi tietojärjestelmien myös organisaatioiden välistä tehokasta tiedonvälitystä. Integraatoratkaisuksi yritysten väliseen viestintään on olemassa useita vaihtoehtoja. Kolmannen osapuolen tarjoama väliohjelmisto ulkoistaa integraation toteutuksen ja vähentää teknisen ylläpidon vaatimuksia, sillä tarjolla on yksi rajapinta, jonka kautta järjestelmä keskustele kaikille kumppaneille.

Integraatiopalveluntarjoajan on kilpailukyvyn ylläpitämiseksi kehitettävä jatkuvasti järjestelmää. Suuret tekniset ja organisatoriset muutokset voivat aiheuttaa tarpeen mittaville päivityksille, joissa asiakasjärjestelmien käytössä olevia ratkaisuja pitää siirtää integraatio-ohjelmistosta toiseen.

Tämän diplomityön tavoitteena on kehittää projektisuunnitelma datamigraatiolle, jossa siirretään edellisen sukupolven integraatio-ohjelmistosta konfiguraatiodataa seuraavan sukupolven integraatio-ohjelmistoon. Datamigraatiolla tarkoitetaan oikea-aikaista datan valintaa, valmistelua, purkua, muuntamista ja pysyvää siirtoa varmistaen riittävä laatu.

Huolellinen migraatiostrategian määrittely ja projektisuunnitelma vähentävät myöhemmissä vaiheissa ilmaantuvia ongelmia. Migraatioprojekti voidaan hajottaa useaan eri moduuliin, jotka voidaan puolestaan jakaa erikseen teknisten resurssien ja bisnesresurssien vastuualueelle. Suuri osa datamigraatioprojektiin liittyvästä työstä on valmistelemaa, ja varsinainen datan siirto ja muuntaminen ovat pieni osa projektia.

Resurssien saatavuus on ennakoitu mittavan migraatioprojektin ongelmaksi. Migraatioprojekti ei todennäköisesti lisää liikevaihtoa, jonka tuottamiseen bisnespuolen resurssit on vahvasti sitoutettu. On myös haastavaa hankkia teknisiä resursseja, jotka tuntevat molemmat järjestelmät ja jotka voivat sitoutua projektiin riittävällä tasolla. Datamigraatio ei ole pääasiassa tekninen ongelma vaan bisnesongelma.

ABSTRACT

PETTERI LEHTINEN: Data migration to a next generation integration middleware

Tampere university of Technology

Master of Science Thesis, 46 pages

May 2016

Master's Degree Programme in Information Technology

Major: Software Engineering

Examiner: Professor Tommi Mikkonen

Keywords: enterprise applications, integration middleware, data migration

Enterprises use information systems as a part of their critical business processes. Integrating the enterprise applications in an efficient way improves the business efficiency and competitiveness. Business processes require intra- and inter-organizational communication. There are multiple integration solutions to overcome these challenges. A third party integration middleware outsources the technical solution making the maintenance easier, because there is only one interface used to communicate with multiple business partners.

Integration provider needs to stay competitive and constantly develop their solution. Major technical and organizational changes may create a need for significant system upgrades and customer solutions need to be migrated between integration systems.

The goal of this thesis is to create a project plan for a data migration to a next generation integration middleware from the legacy system. Data migration means the selection, preparation, extraction, transformation and permanent movement of appropriate data that is of the right quality to the right place at the right time and the decommissioning of legacy data store.

Proper migration project plan and strategy prevent future problems. The project is broken down into multiple modules that divide the responsibilities of the business and technical sides of the project. A significant portion of the project happens before any actual data transfer and transformation.

Availability of resources is seen as major challenge in the project. The migration project is unlikely to generate any revenue, and that is where the business resources are committed to. It is also challenging to engage technical resources that are familiar with both the legacy and next generation systems. Still, data migration is a business issue not a technical one.

PREFACE

This master's thesis is examined in Tampere University of Technology. This thesis contains a project plan for a data migration between two integration systems. The plan is presented in high level containing tools and techniques in managing the process, and technical details are left out partly to avoid disclosing critical business information.

I would like to thank Professor Tommi Mikkonen for helping me find the best way bring this project to a finish. Special thanks go to Mikko Javanainen and Lauri Hölttä for guiding the creative process and helping out in the complex world of integration, and to everybody else taking part in evaluating the ideas.

I'd like to thank my family and friends for supporting me and pushing me to finish this project, not least because of the peer pressure.

In Tampere,

Petteri Lehtinen

CONTENT

1	Introduction	1
2	Enterprise application integration	2
2.1	Enterprise application characteristics	2
2.2	Motivation and challenges in business process integration	2
2.3	Different approaches to integration	4
2.4	Message-oriented integration.....	8
3	Data migration on enterprise applications	12
3.1	Drivers and requirements.....	12
3.2	Basic processes and common pitfalls	13
3.3	Data migration strategy.....	15
3.4	Data migration project modules.....	16
3.5	Tools	18
4	Configuration data migration between integration middleware systems.....	21
4.1	Characteristics and limitations on integration middleware	21
4.2	Metadata modelling and key business data area decomposition.....	21
4.3	Key data stakeholder management	22
4.4	System retirement plan	24
4.5	Data profiling and landscape analysis.....	25
4.6	Managing and enforcing data quality rules.....	26
4.7	Gap analysis and mapping	28
4.8	Migration end-to-end design.....	29
4.9	Software selection.....	30
5	Data migration design for a messaging process	32
5.1	Migration strategy.....	32
5.2	Conceptual entities.....	33
5.3	Target model	35
5.4	Conceptual entity mapping	37
5.5	Migration end-to-end design.....	38
5.6	Go-live monitoring and fallback readiness	40
6	Evaluation of the migration design	42
6.1	Peer review	42
6.2	Architecture review	43
6.3	Future work.....	44
7	Conclusions	45
	References.....	47

1 INTRODUCTION

Enterprises use information systems as a part of their critical business processes. Integrating the enterprise applications using a third party integration provider improves the business efficiency and competitiveness, while providing a single interface to communicate with business partners. Integration providers need to stay competitive and develop their solutions. Major changes in the solution may create a need for significant system upgrades and customer solutions need to be migrated between integration systems.

The large scope of this thesis is data migration between two integration middleware: an existing middleware with extensive user base and data structure, and a new middleware solution being developed while writing this thesis. The main objective is to create a process for migrating configuration data from the existing middleware on to the new one.

Quality assurance perspective needs to be addressed, and providing a reliable fallback option is an important goal. The process is planned from a data-centric point of view, which means that technical details and possible organizational requirements and corporate politics are discussed only if they have major implications to the project.

This project takes regard only configuration data, which is defined in the terms of these two middleware. Other types of data are presented when needed to give understanding on the use of configuration data or quality assurance. Integration middleware and organizational structure are introduced in the scope of the migration process and configuration data use when necessary.

One of the goals is to create a general concept of a reliable configuration data migration process rather than make a particular case about the middleware used as a framework for the thesis. However, a complete generalization is infeasible as middleware characteristics define parts of the process itself.

Chapter 2 takes a deeper look on enterprise applications and integration, and different integration styles. Chapter 3 discusses data migration to give a general background for the process plan. Chapter 4 defines data migration tools and techniques. Chapter 5 presents a process plan for configuration data migration on to the target system. In Chapter 6 the process is put under peer and architectural reviews. In Chapter 7 the research process is summarized and evaluated.

2 ENTERPRISE APPLICATION INTEGRATION

2.1 Enterprise application characteristics

The term *enterprise application* is used to describe a software system operating in a corporate environment. Examples of such applications include *enterprise resource planning*, *supply chain management*, *customer relationship management* and *electronic commerce systems* that enable organizations to improve their focus of using information systems to supporting their operational and financial goals (*Puschmann & Alt, 2001*).

Global business operations need enterprise applications to integrate enterprises in a supply chain environment to achieve competency and competitiveness. Enterprises like Dell and Microsoft have adopted enterprise resource planning systems in order to profit of a global supply network. Smaller companies are also quickly learning that a highly integrated enterprise application system is a requirement for the global operation. It is well recognized that enterprise applications have a great impact on global industrial development. (*Fowler, 2002*)

Business organizations are flexible networks that have intra-organizational and inter-organizational relationships between business units. Efficient coordination between these units can only be achieved by delivering exact and right information for different business functions throughout the supply chain. (*Puschmann & Alt, 2001*)

In typical organizations the business functions are spread across multiple applications that focus on specific core functions. However, it is difficult to provide a clear separation between the functions and often there is a need to share information between applications designed for different functions (*Hohpe & Bobby, 2006*).

Enterprise applications are usually involved with a lot of data stored for several years. During this time the applications, operating systems, and hardware using the data are bound to change. Often there is a high level of concurrency as the data is accessed by multiple users at the same time. (*Fowler, 2002*)

2.2 Motivation and challenges in business process integration

Growth of Internet, business expansions, competitive pressures, new business models, and the need to optimize business processes internally and with business partners have made business integration an important topic for many companies. Typically term

enterprise application integration is used as a catch-all concept to cover all the aspects of business integration. (Qureshi, 2005)

So, on a high level enterprise application integration is a set of plans, methods and tools to connect, unify and coordinate business functions (Xu, 2011). From software point of view, an enterprise has existing systems that are expected to stay in service while new features are being added or migrated to a new set of applications (Xu, 2011). In this thesis the term *enterprise application integration* is used in the scope of software solutions that can help to achieve better business integration inside enterprises and with their partners and suppliers.

Enterprise application integration is an attractive proposition to businesses because it offers an unrestricted way to share information between all connected functions and to present a seamless chain of processes to business partners (Erasala, 2003). So, the principal motive for enterprise application integration is to achieve greater automation, which eventually improves overall business efficiency (Hosseing & Ardeshir, 2013).

Enterprise application integration provides cross-platform, cross-language integration between the business applications allowing them to share messages and data (Hohpe & Bobby, 2006). Enterprise application often incorporates an n-tier architecture enabling it to be distributed across several computers (Hohpe & Bobby, 2006). A main software architecture for n-tier enterprise applications have been established, where web servers, application servers, and database servers run on their own tiers (Calton Pu, 2011).

In n-tier architecture one tier may not be able to run by itself (e.g. data tier application running queries to the database), but enterprise applications are independent programs that can each run by itself. Application running on multiple computers and tiers alone is not considered application integration but application distribution. Distributed applications tend communicate synchronously and often they have human users that only will accept near real-time response. (Hohpe & Bobby, 2006)

The difference application integration makes to application distribution is that communication is usually asynchronous and time constraints are not too restricted, because immediate response is not expected. Another defining concept is the measure of interdependence between application components, called coupling. If components depend on few details of each other they are called loosely coupled. (Hohpe & Bobby, 2006)

Integrated applications usually communicate in a loosely coupled way and they each run by themselves. This method of design enables each application to focus on a definite set of functions and to delegate to other applications for related functionality. Integrated applications typically have broad time constraints making asynchronous work possible. (Hohpe & Bobby, 2006)

A self-sustained integration system is needed to move messages from one computer to another because computers and networks are inherently unreliable. Just because one application is ready to send data does not mean that the other application is ready to receive it. Even if both applications are ready, the network may not be working, or it may fail to transmit the data properly. Integration system overcomes these limitations by repeatedly trying to transmit the message until it succeeds. Under ideal circumstances, the message is transmitted successfully on the first try, but circumstances are often not ideal. (*Hohpe & Bobby, 2006*)

Business-to-business or inter-organizational integration has a long history in integration world, having existed at least as long as the Internet (*Medjahed, et al., 2003*). This kind of integration emphasizes the existing challenges and presents some additional problems compared to intra-organizational integration. Business partners' geographical locations may be wide apart factoring to network reliability, the information systems may be vastly different and continuously developed for different purposes, and interpersonal communication between two organizations is likely more time-consuming and less transparent compared to discussions inside an organization (*Hohpe & Bobby, 2006*).

Companies have tried to overcome some of the integration challenges by simply focusing on a single application provider. This has not worked as well as expected. One software provider is not able to cater for all the organization's requirements, and there is often a need for integration with other business partners with a different variety of software. In inter-organizational enterprise application integration homogenous architectures are not a practical option (*Thomas Puschmann, 2001*).

2.3 Different approaches to integration

To manage all the complexities in business integration, appropriate *integration middleware* technologies are required. Integration middleware is essentially a software system that is capable of interfacing with two or more different applications and enabling reliable and efficient communication between them (*Hohpe & Bobby, 2006*).

Basic physiology for these middleware solutions varies. On enterprise level one classification is EDI based solutions, component based message-oriented middleware, workflows, XML frameworks, and web services. EDI provides inter-organizational transfer of business documents in a compact and standard form. EDI defines single solution for content interoperability and is hardly flexible in its ability to expand the supported document types. (*Qureshi, 2005*)

Component middleware is a framework based on interconnected distributed components. The components can be created, deployed, and the interaction among them is

configurable. Components are modules that can be independently developed from scratch or created by wrapping existing functionalities in them. (Qureshi, 2005)

In an application, business process consists of business related activities formed by data and control flow relations. Workflow management deals with declarative definition, enactment, administration, and monitoring of the processes. Workflow technology is important for automating business processes involving access to multiple applications. (Qureshi, 2005)

XML based approach allows the use of services without mediation facilities and dedicated transformation or specific integration of partner's system. Business processes can be integrated in terms of agreed upon documents and a common XML schema eliminates dedicated translation of information. So, effective integration requires standardized schemas, mapping, processing, and service invocation. The issue of interoperability shifts from the applications to the level of standards. (Qureshi, 2005)

Web services are loosely coupled applications using open, cross-platform standard and they operate across organizational and trust boundaries; they are free from client context and deployment requirements. They are dynamically located and invoked, firewall friendly and widely accessible which makes them appropriate for economical enterprise application integration solution. (Qureshi, 2005)

There are four main technical approaches the integration challenges have been overcome by developers:

- **File transfer** is an approach where one application writes a file to a location accessible by another application that later reads it. File name, the format and which application will delete the file need to be agreed.
- **Shared database** refers to solution where all applications use a single database so no data has to be transferred between applications.
- **Remote procedure invocation** is a process where an application accesses other application's public procedure in real time and synchronously.
- **Messaging** is an asynchronous approach, in which applications publish a message on a channel wherefrom other applications can read it. (Hohpe & Bobby, 2006)

In the scope of this thesis messaging is the approach to be discussed in more depth. However, a complex integration middleware may be a combination of many of these approaches.

Table 1 collects common features and requirements for a functional integration middleware solution. Each concept provides additional description regarding how the requirement affects the integration middleware and businesses.

Table 1: Feature requirements for enterprise integration application solutions (Adapted from Qureshi, 2005; Chung & Leite, 2009).

Architectural requirements	
Scalability	System's ability to grow in different ways in a low cost manner. For example adding new trading partners for businesses and creating new connections should require low effort.
Security	System should take sufficient actions to meet at least the basic information security requirements. Additionally any business partners' required security requirements need to be met accordingly.
Heterogeneity	Between different connected parties there may be a high level of heterogeneity – structural and semantic. These challenges need to be overcome by for example message transformations.
Adaptability	It should be possible to quickly adapt to business changes.
Manageability	System needs to provide sufficient services for tracking, notification, configuration and recovery.
Distributivity	Different services should be separable to different layers or integration levels and manageable so that the required level of integration is achieved.
Autonomy	More local control over services and processes. Ability to change processes without affecting each other.
Decoupling	Connected partners should not be depended on each other but rather be able to exchange business information on demand.
Real-time requirements	
Asynchronous	When business demands, it should be possible to send information without the expectation of an immediate response.
Publish/subscribe	Provides many-to-many interaction where only one message needs to be produced and it can reach multiple recipients.
Business requirements	
Flexibility, agility	Possibility to change processes according to business logics while supporting system autonomy.
Usability	Human factors, aesthetics, consistency, documentation.
Reliability	Involves the availability of components and integrity of information maintained and supplied to and from the system.

While many of these concepts as such are desirable in many kinds of software systems, integration middleware has properties that make certain attributes critical for successful operation. Integrating business functions emphasizes the business logic requirements that resonate to architectural features. Sufficient security becomes crucial, adaptability,

heterogeneity, and manageability are important for changing business logics and all the while usability and reliability requirements need to be met. Often, practically uninterrupted availability is required.

Hosseing & Ardeshir have developed a proposed list of integration application core features of which a specific set is usually implemented in an integration middleware, depending on what kind of problems the technology is designed to solve. (*Hosseing & Ardeshir, 2013, ss. 21-24.*)

Table 2: Integration application core functionalities and supporting features. (Adapted from Hosseing & Ardeshir, 2013, ss. 21-24.)

Core integration functionalities	
Messaging	Data transfer between applications.
Routing	Distributes messages from application to one or multiple other applications.
Persistence	Stores the exchanged data and related events.
Connectivity	Shares information and application services between client applications.
Transformation	Supports conversion and transformation of exchanged data.
Transaction management	Support for long and short term transactions.
Business semantics and metadata	Support for business semantics as well as metadata (message dictionary) and metadata representation service.
Business Rule Management	Support for the management of business rules and rule engine.
Process Management	Support for the orchestration, coordination and management of fully-automated business processes where the composition of process elements occurs fairly rapid.
Bulk Data Movement	Support for high-performance and bulk data movement between two data sources.
Human Workflow Management	Support for the coordination and management of semi-automated workflows where the composition of process elements occurs in long-term.
Activity monitoring and event management	Support for the monitoring and supervising on business processes.
Partner management	Support for inter-enterprise process coordination as well as B2B integration features.
Supporting integration features	
Process Modeling	Support for modeling, optimization and abstraction of business processes.

Process Simulation	Support for the simulation of business processes.
Development and Support	Support for the development services such as transformation specification and interface development along with extensibility, modifiability, supportability and testability of integration solution.
Administrative and Runtime	Support for the administrative and runtime services such as distribution, scalability and workload balancing, monitoring, performance and availability (recoverability, reliability and failure handling).
Ease of Integration	Simplicity or complexity of integration tools through off-the-shelf components to facilitate legacy and mainframe integration and integration without programming as well as support for non-invasive integration.
Flexibility and Portability	Flexibility in functionality and modification along with the portability of integration solution (topology and platform independence).
Security	Support for the security of integration solution.

This thesis focuses on technical features and challenges in integration middleware. Each technology choice and feature set implementation will provide its own set of challenges. Even though all core integration features listed in *Table 2* are present in some way, this thesis discusses only a limited set.

2.4 Message-oriented integration

Messaging is a type of communication technology designed to meet the requirements set to an integration solution. Messaging is a reliable, high-speed, asynchronous communication technology between applications. A messaging system is configured with different channels defining the paths of communication between the connected applications. The messaging system then coordinates the reliable sending and receiving of messages. (*Hohpe & Bobby, 2006*)

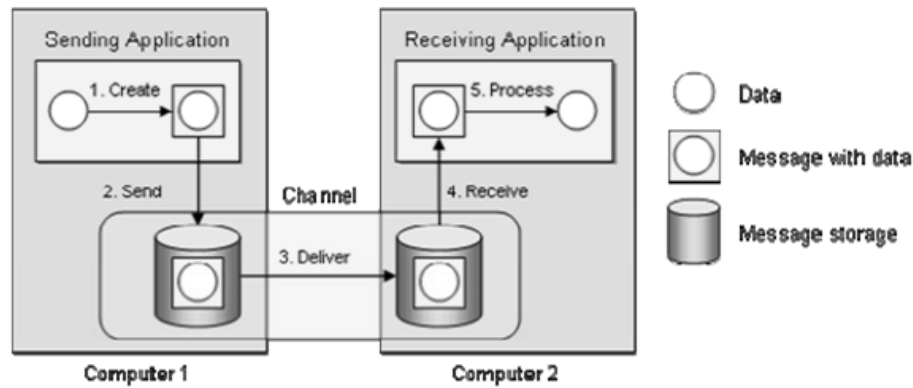


Figure 1: Transmissions steps in a messaging system. (Hohpe & Bobby, 2006)

Figure 1 above is a crude overview of a messaging system's transmission steps. Some of the root patterns in messaging are also present and they are expanded below in table 3.

Table 3: Root patterns in messaging. (Hohpe & Bobby, 2006, pp. 75-108)

Design pattern	Description
Message	Messages are essentially data structures that can be interpreted as data, a command or information about an event in the sending system. If the sending application is expecting a response the message should provide an address for the response channel. Huge amounts of transfer data and slow messages pose more challenges to messaging system as well as the connected applications.
Channel	Messages are delivered through channels. A channel is a collection of messages that can be accessed by all connected programs. A sender program writes messages on to these channels and a receiver reads and usually deletes it from the channel. Programs can act as both, a sender and a receiver.
Pipes and filters	<i>Pipes and filters</i> is an architectural style that allows easy combination of individual patterns into a larger solution. <i>Pipes</i> are messaging channels that chain together <i>filters</i> that each are a single processing step in the integration process, e.g. message translation.
Router	Message routing is used to route messages from one inbound channel to one or more outbound channels. More complex flows may be created by combining multiple simple routers. Content-based routing inspects the content of a message and routes it to another channel based by predefined rules.

Translator	A message translator converts the message into another format so it can be used in a different context. Translation can happen on different layers: data structures (entities, associations), data types (field names, constraints, value domains, code values), data representation (data format, character set, encryption), transport (TCP/IP, http, SOAP)
Endpoint	Message endpoint is connected to a messaging channel. It can then send or receive messages. The endpoint encapsulates the messaging system from the rest of the application and customizes a general API for the task.

Even though asynchronous messaging solves many of the problems in an elegant way it also introduces new challenges. Some of these challenges are inherent in the asynchronous model while other challenges vary with the specific implementation of a messaging system. *Table 4* below lists most common challenges faced in a messaging solution.

Table 4: Challenges in asynchronous messaging. (Hohpe & Bobby, 2006, p. 18)

Challenge	Description
Complex programming model	Application logic can no longer be coded in a single method that invokes other methods, but rather it is now split up into event handlers responding to messages in the channel. Development process for such a system is more complex compared to direct method calls.
Sequence issues	Message channels guarantee that the message will be delivered, but they do not guarantee when the message will be delivered. This can cause that messages sent earlier in time will be received later than messages sent afterwards.
Synchronous scenarios	Not all applications can operate in a send and forget mode and there may be cases where real-time response is required.
Performance	Messaging systems add some overhead. It takes effort to make data into a message and send it, and to receive a message and process it. Messaging is often not be the optimal solution to send huge chunks of data.
Limited platform support	Many proprietary messaging systems are not available on all platforms. Often times it is easier to FTP a file to another platform than accessing it via a messaging system.
Vendor lock-in	Messaging system implementations rely on non-standard protocols. As a result, different messaging systems usually do not connect to one another.

A company offering integration services may have multiple integration solutions actively in use, under maintenance, and under development. This may have happened due to business acquisitions or introducing a new version integration solution. There may be a need for these systems to communicate with each other, but there may also be a motivator to run down the existing solutions and fully utilize a single system. This introduces an attractive opportunity for a migration of the existing data from the legacy system to the new one.

3 DATA MIGRATION ON ENTERPRISE APPLICATIONS

3.1 Drivers and requirements

Data migration is discussed in terms of enterprise application migration, meaning the one-off movement of data from old system to a new repository. Data migration can be defined as “*the selection, preparation, extraction, transformation and permanent movement of appropriate data that is of the right quality to the right place at the right time and the decommissioning of legacy data store*”. (Morris, 2012, p. 7)

Due to business consolidation, mergers and acquisitions, companies working more on a cross-functional basis and less in silos, there is a need to simplify the data storage systems architecture, to create reference data and centralize them in a unique repository. (*Clement Dephine, 2010*)

Like with enterprise applications, during an integration middleware’s lifetime a lot of data is stored and persisted for a long time. The different integration features are configured to support multiple client applications and communicate with the clients. This configuration data is used to automate transformations and routing performed for the incoming messages sent by client applications. The configuration data may be specified on a level of single connection or message type, and some of it needs to be manually created and maintained. (Hölttä & Javanainen, 2016)

Competitive businesses and their business applications are constantly evolving and middleware technologies are not an exception. Integrated applications demand more performance, capacity requirements are getting higher and more business functions are being automated creating even more demand for integration. Old middleware solutions may not be able to meet all the requirements and in a few years’ time they may become obsolete. Therefore, integration middleware needs to be upgraded to support new technologies, and sometimes heavy changes are required to core features and their design affecting also configuration data format and requirements. These changes create a need to transform the existing configuration data some way or even migrate to a new middleware. (Hölttä & Javanainen, 2016)

The source middleware system for the data migration processes is referred as *legacy middleware* and is the host of all legacy data stores. The data migration project’s goal is to move a set of data to a new data store. The system utilizing the migrated data is referred as *target middleware*.

3.2 Basic processes and common pitfalls

There are different forms of migration and the decision of the form will affect other decision further in line. The most common forms are listed in *Table 5*. A migration project may be a combination of these different forms and it might change in the lifetime of the project.

Table 5: Common forms of data migration projects. (Morris, 2012, p. 54)

Migration form	Explanation
Big bang	All data is moved at once and legacy systems is decommissioned.
Phased	The data is moved in separate parts.
Parallel	The data is moved to the target and the legacy system is allowed to run alongside it for a period of time.
Always up	Migration where the source system cannot be taken offline at all.

Nearly 40 per cent of data migrations projects are not finished on schedule or budget, or they fail completely (Howard, 2011). The reasons for failure are diverse. If the scale of required activities is not well known it may lead to underestimating, which is especially true for estimating the amount of data preparation activities required (Morris, 2012, p. 9).

Seeing data migration as a purely technical problem is one of the causes for failed migration projects. In fact, the migration technology itself is rarely a problem. Rather, successful migration projects put business engagement ahead of other success criteria. Still sufficient specialist skills are required and the migration project team needs to be able to understand the business and technical processes and provide each with proper leadership. (Morris, 2012, p. 9)

There will be data issues in the legacy system. Some of them are well known and some of the not know or forgotten. Some may be known but no one is willing to share them with the technical team. (Morris, 2012, p. 19)

Responsibility gap is the biggest cause for failing migration projects. The principal reason for this is a situation where data quality issues swirl between the technical side and a disengaged business side. Each side is expecting the other the fix the issues in a confusion of roles. The best way to avoid a responsibility gap would be avoiding it in the first place and creating a project where all parties are working in collaboration. (Morris, 2012, p. 15)

Creating a collaborative project team where all stakeholders are engaged into bringing migration process to the successful finish requires certain mindset and precautions. From the common reasons to failure and tried and tested methodologies, four most important

rules for a successful migration project are:

- **Data migration is a business not a technical issue.** Data migration is normally a result of an IT project and IT projects are there to answer a business need. Business understands the meaning and value of the data and the cleansing, preparation, extract and transform operations must be derived from this understanding. *Business must own the quality of the data and success of the migration outcome.* (Morris, 2012, pp. 17-18, 20)
- **The business knows best.** The business knows where the important legacy data sources are located. The business has the expertise to make judgements about data quality and appropriateness, the technical team is there only to facilitate the process. *The business takes the ownership of the decisions made in the process and the ownership of the results.* (Morris, 2012, pp. 23-24)
- **Data will not be of perfect quality.** Data quality compromise is the norm. This happens due to compromises made in the schedule and budget. That combined with breaking previous rules the technical side will seem to have failed to deliver their promise and will be perceived out of touch with the needs of the enterprise. From technical point of view the business has failed to communicate their requirements appropriately and on time. *Technical-business relationship needs to be structured so that the business decides the priorities – early on.* (Morris, 2012, pp. 25-26)
- **Define measurable migration items.** There needs to be a way to measure achievements and remaining issues. The measured items should be meaningful to the enterprise and the migration readiness needs to be communicated in business terms. There usually is a difference how one migration item is perceived in technical point of view and in business terms. In technical perspective it may mean a single row in a database but for a business it is the smallest thing the business has interest in – for example a customer. *The migration items need to be decided between the project and the key stakeholders.* (Morris, 2012, pp. 26-27)

Demilitarized zone is a method to ensure the responsibilities and exchange between the business and technical resources are clear and understood. The demilitarized zone allows clear separation of the different roles in the migration process. The technical resources can hide the work of the business resources and vice versa. The overlapping area of the resources would be the demilitarized zone. (Morris, 2012, pp. 22-23)

Furthermore, there will be challenges in building the team required for a successful project. People will resist change, and convincing them to work within the assigned tasks is not always easy.

3.3 Data migration strategy

Many of the pitfalls in migration projects are a lot easier to avoid at the outset than fix after starting the project. Data migration strategy states what is to be done, how the project partners are interfacing, and what controls are put in place to properly manage the activities. Data migration strategy is an essential document whose delivery is normally led by the project manager, but it will involve all the senior management in decisions about scope, reporting and strategic guidance. *Table 6* lists a set of project items the data migration strategy documents should address. (Morris, 2012, p. 46)

Table 6: Data migration strategy should address following project items. (Morris, 2012, pp. 47-74)

Project item	Content and actions
Project overview	A high level explanation of the process, written for someone not necessarily familiar with migration methodology.
Project scope	Defines which legacy data stores are migrated and which areas of functionality are focused on. The scope often changes during the lifetime of the project.
Budget	The data migration project should have its own budget. However, normally a more accurate budget can be generated only after the initial analysis of current state and how much it will cost the make the change to the desired state. Normally this analysis will have a more accurate budget and the initial budget for rest of the project is indicative.
Project and programme organization	This thesis does not describe the constitution of the programme organization, but it is expected that some industry standard pattern is followed. The migration project does not usually justify its own project board, but the project manager should have access to the bigger project programme's board.
Modules	Describes which modules (overview in chapter 3.4) are used in the project.
Policies	Explain business drivers and constraints affecting the project and project environment. Policies include project methodology, architectural requirements and risk aversion principles. Master data management should have an own policy that should indicate whether master data should be managed centrally. This part should contain all regulations the organization is subjected to and also any

	local policies the organization wants to impose on the project.
A project decomposition	Describes how the project is broken into manageable pieces. For example, is it broken down data-centrally by different data stores or by different business units in the organization?
The form of the migration	Initial decision on the form the project will take. Most common forms are listed in Table 5.
An initial migration plan	A high level plan including tasks according to the project modules. Should include the go-live date for target system.
Software selection	A list of data migration software to be used in the project, including data proofing and quality tools and migration controller. The budget and scale of the project, as well as migration from and local preferences will affect this.
Initial lists	Already in the beginning there will be some initial legacy data stores, key data stakeholders and data quality rules the project organization is aware of.

As this thesis focuses on configuration data migration master data management activities will have a smaller role.

3.4 Data migration project modules

The migration project can be broken down into different modules that help tailoring it to different delivery methods. There is some recursion in the modules during the project, for example when new requirements emerge. *Figure 2* presents high level view on the modules to be used in the migration project. (Morris, 2012, p. 35)

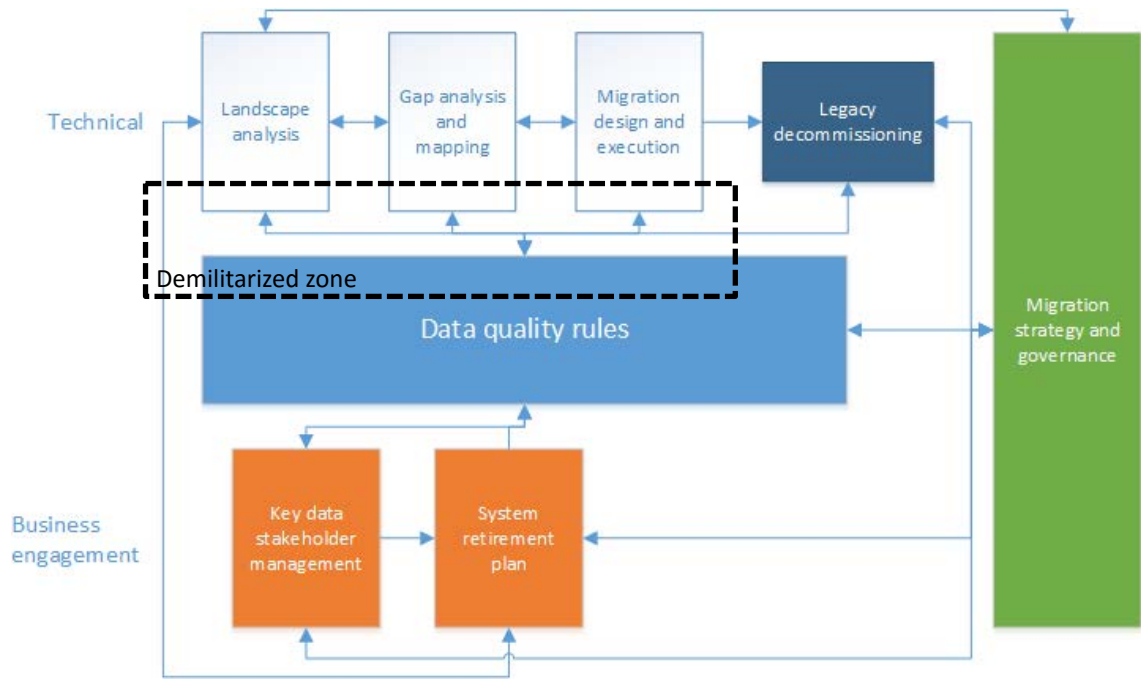


Figure 2: Data migration modules and product flows. (Morris, 2012, p. 35)

As seen in Figure 2 there are two types of functional modules: *technical* and *business engagement*. It's important that they both are integrated within the project. Data quality rules span both of these. Each module is briefly explained in Table 7.

Table 7: Project modules explained. (Morris, 2012, pp. 36-40)

Project module	Goals and activities
Landscape analysis	The goal is to discover and catalogue legacy system's data stores and their relationships. This includes also non-official data stores that may be hidden away to departmental desktops. Manual and automated data profiling is done in this point to find out what kind of data is stored and what data challenges there might be. Landscape analysis may also be used as a fast iteration to first quantify the scale of the migration task prior setting the budget. Landscape analysis may be started already before the target system is well defined.
Gap analysis and mapping	Data mapping means linking the fields in the legacy data store to the ones in the target system and defining the required data transformations. Any gaps between the systems will be documented and managed in this module regardless in which module the gap was found.
Migration design and execution	Migration design and execution phase is where the design, test and execution of migration and archiving are carried out, also taking business limitations, timings, audit requirements, data

	lineage, fallout, fallback, reporting, management and control in to account.
Legacy decommissioning	Legacy decommissioning covers the physical or logical removal of legacy databases, hardware and software.
Data quality rules	Data quality rules are central part of the data migration project. It relates to all data quality and data preparation related activity in the project. It involves the technical experts from the legacy and target systems plus the business domain experts allowing them to collaboratively prioritize, manage and complete data issues.
Key data stakeholder management	Manages the discovery, briefing and management of these individuals. There is as many business as technical roles, most important ones being data owners and business domain experts. Data owners are defined as “all the people within or outside an organization who have the legitimate power to stop a migration from happening.”
System retirement plan	Data migration project’s ultimate goal is to turn off all legacy data stores. System retirement plan is a key input to migration design and execution, and it allows the project team to be confident with the migration by asking series of structured questions to elicit the business view of the migration and to make sure requirements are constantly met allowing an eventual legacy decommission.
Migration strategy and governance	Covers standard programme management functions expected on a well-managed project. Includes the creation of a data migration strategy.
Demilitarized zone	The interface between the technical roles and responsibilities of the business roles. Key component of the migration project and to be defined formally.

The modules in *Table 7* represent the high level softer issues in the migration project, but in them there are processes requiring specialiced tools and appropriate software support.

3.5 Tools

Bloor Research has identified three best practices that include tools and methods used during data migration projects (*Howard, Data Migration, 2011*):

- **Data profiling tools** should be used before and during the project. Data profiling uncovers data quality errors, relationships, discovers sensitive data and monitors data quality on an ongoing basis. Data profiling should be executed before setting budgets and deadlines because profiling enables identification of the scale of the issues that may be involved.

- **Data quality tools** can improve data quality and is not a process to be done manually. Data quality is used to assure accurate data and to enrich it.
- **Migration controller** will be needed in order to move the data and this should also have the collaborative and reuse capabilities.

These tools also represent the different phases of the project where technology support is available. In *Figure 3* the modules are changed to contain appropriate tools: profiling tool, data quality tool and migration controller.

The first step in landscape analysis is to find all legacy data stores with any data of interest. The most basic tool for discovering these is to publish a visible amnesty on finding the private data stores around the organization. This is meant to be as inclusive as possible as key data stores should not be overlooked here, still taking the quality versus time budget in account. (*Morris, 2012, p. 106-107*)

Additionally, there are multiple existing software solutions supporting the tasks in landscape analysis phase. During landscape analysis software can provide support for profiling data stores. Profiling tools discover the rules in the legacy data stores and uncover other unforeseen features at column and row level. Profiling is used to build an informative list of the legacy data stores including potential anomalies. (*Morris, 2012, p. 41, 125*)

During landscape analysis also data quality issues may arise. Quality issues at this point often relate to the master data as during profiling it becomes apparent that legacy data stores contain duplicate information. It must be decided within data quality rules how much effort will be put in to the master data deduplication. (*Morris, 2012, p.120-124*)

The data quality issues uncovered throughout the project are referred as data gaps. They are managed inside the *gap analysis and mapping* module regardless in which phase they arise. Data quality tools are used in the gap analysis and mapping module to uncover these data gaps once the project has an idea of the target. Data quality software should allow implementation of validation and cleaning rules. There is overlap between profiling and data quality tools, but in general profiling tools are for discovering the relations and possible quality issues, while data quality tools enforce the known data quality rules. (*Morris, 2012, p. 42*)

Migration controller fits in the *migration design and execution* module (*Figure 3*). Migration controllers are known as the extract, transform and load tools, but they are expected to do more than performing just these functions. Migration controller is responsible for reading the data from legacy data store, validating the extracted data (preferably using the data quality tool), reformatting and blending the data from multiple sources, scheduling the migration process and writing the data to the target, while managing fallout or fall back and reporting on execution and fallout, and providing audits. (*Morris, 2012, pp. 42-43*)

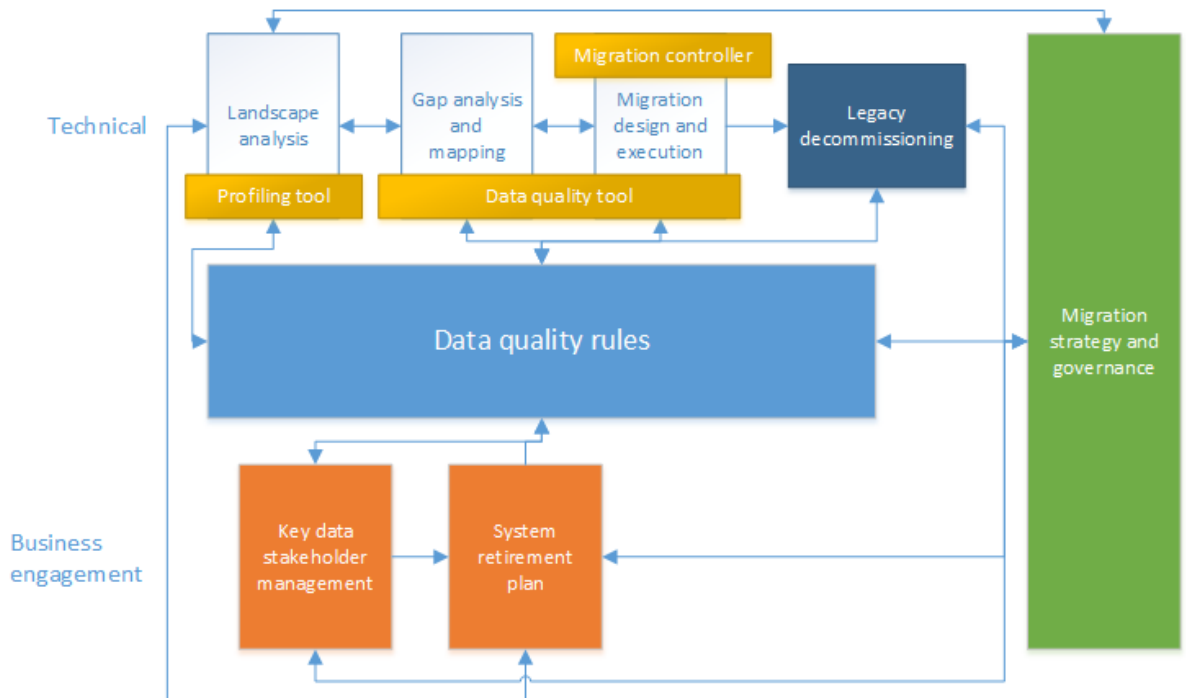


Figure 3: Data migration tools within the programme. (Morris, 2012, p. 35)

As a part of the *migration design and execution* module an end-to-end design is provided that presents an architectural overview describing the migration from the first extract of data to the last legacy data store decommission. The end-to-end design should define the data transport, migration, transitional data store, master data management and archive software to be used during the execution. (Morris, 2012, p. 178)

As seen in Figure 3 *data quality rules* are closely related to each of these modules. Data quality rules are processes and deliverables that can measure the data quality within the migration project. The definition of the tool is intentionally loose as quality can be defined techno-centrally as well as by business rules. Data quality rules are defined anytime during the data migration project but preferably in the earlier phases. All data quality rules should be stored in a single location that allows forming a single picture of current state of data readiness and acceptability. Data quality rule management is better described in chapter 4.6. (Morris, 2012, pp. 140-141)

4 CONFIGURATION DATA MIGRATION BETWEEN INTEGRATION MIDDLEWARE SYSTEMS

4.1 Characteristics and limitations on integration middleware

A well-established middleware technology used for critical business functions cannot begin to malfunction nor can it be offline for any prolonged time. This makes quality assurance and emergency fall back methods a vital factor for the migration process. Data quality is essential for a successful migration project and relevant especially for the preparation, extraction and transformation phases (*Clement Dephine, 2010*).

The data migration process plan will focus on configuration data migration, and master data management or transactional data migration will be left out. Attention need to be paid for processes happening after the target system go-live, including system monitoring for quality assurance.

4.2 Metadata modelling and key business data area decomposition

Metadata modelling provides a level of abstraction that enables comparing multiple different data stores. The four models to be used include the conceptual entity model, the migration model, the target model and individual legacy data store models. When target model is defined, it can be used as the migration model. (*Morris, 2012, p. 79*)

The conceptual entity model groups atomic entities together so that they are meaningful to the enterprise, so one entity is a unit of migration that is valuable and meaningful to the organization and its functions. Data store models are models of each individual legacy data store some of which are usually known already in the beginning phase. Landscape analysis will bring up more as the project proceeds. (*Morris, 2012, p. 82*)

In messaging middleware systems, the conceptual entities related to configuration data will consist of messaging system design patterns. Each of them can make up a unit of migration. Together the conceptual entities, such as a channel, router and translator, form a *pipes and filters* architecture. *Figure 4* illustrates the relations of these entities.

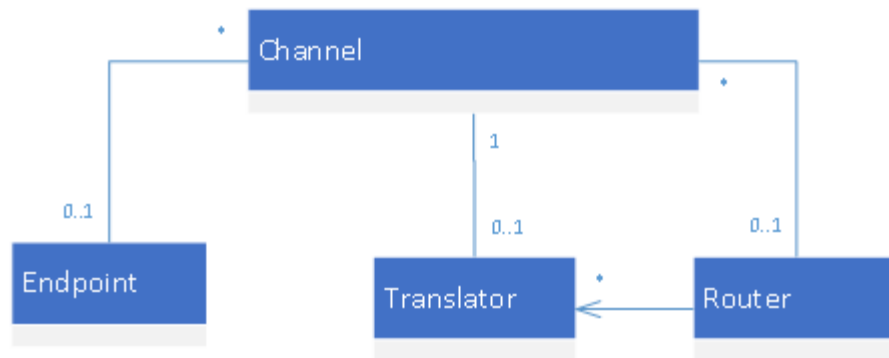


Figure 4: Conceptual entity relationships and multiplicity in messaging design patterns.

In Figure 4 channel pattern presents input and output channels. Endpoint related to an input channel is the entrance point to the messaging system and it may publish to multiple input channels. A channel connects different components together, always having an endpoint, a translator or a router connected. Router may have a translator directly connected and it may publish messages to multiple channels. A translator may contain multiple transformation processes.

Project decomposition can be data-centric or by business function areas. From the data point of view, there are clear separate conceptual entities, and it makes sense to base key business data areas on these. When the target system is modularized the decomposition and data areas may follow the target modules, so each key business data area will have a single expert stakeholder. (Morris, 2012, p. 87)

Data migration form will also affect the key business data area decomposition. For example, with phased delivery it may be reasonable to divide the work by phase. Smaller projects may not even merit any subdivisions. (Morris, 2012, p. 87)

4.3 Key data stakeholder management

A successful data migration project needs to be enterprise-led and ownership needs to be assigned in the beginning before issues start to arise. The enterprise owns the ultimate delivery and the solutions to the problems and it will always have the last sign-off responsibility. A successful alliance needs to be built with the communities within which the knowledge lies. The key data stakeholders will live with the results and they need to be identified. (Morris, 2012, pp. 91-92)

A data stakeholder is any person within or outside an organization who has a legitimate interest in the data migration outcomes. Outcomes refer to the project deliverables, for example successfully migrated business relevant data or an audit trail of the migration processes. There can be stakeholders in multiple different roles, two most important ones for every data store being *data owner* and *business domain expert*. (Morris, 2012, pp. 93-94)

The data stakeholders can act in following roles:

- **Data owner** is the one who can sign off the decommission document and switch of the legacy data store. Often data owner is not someone in day-to-day contact with the data source, but may be a senior manager. They may not exactly know how the data source is used today. Data owners may be a group of people each of whom take responsibility for a part of the legacy data store. (Morris, 2012, p. 94)
- **Business domain expert** is someone who knows how the data source is used currently. Business domain expert and data owner can be the same person, important thing is the degree of up-to-date knowledge. The person may not necessarily work for the data owner. The business domain expert should be readily available to attend workshops and take phone calls. Complex systems will need more than one business domain expert, each familiar with their own aspect of the data source and its use. (Morris, 2012, pp. 95-96)
- **Technical system expert** knows the system specifics that are not implicit in the data store but enforced by the system, such as file formats, access rights and validation. It is likely easiest to find technical system expert out of all data stakeholders. System experts are also well informed on issues of data quality. The expertise may be of wide variety: legacy data store, target data store, connectivity or system topography. (Morris, 2012, pp. 96-97)
- **Data migration analysts** are the people who understand the migration project composition and methodology and are able to adapt the practice to the underlying principles. Data migration analyst is part of the data quality rule board and responsible for ensuring that product quality and timeliness is satisfactory. The analyst bridges the technical, business and programme activities. Data migration analyst should be able to articulate technical issues to business people and vice versa, create right environments for conversations, data architect's skill to understand relationships, project manager's skill to manage time and budget and technologist's skill to understand the available technology and its best use. (Morris, 2012, pp. 97-98)
- **Corporate data architects** are responsible for the design of how the data required for an organization is held. They will have an overview of how the data is structured and where it is mastered in those cases where the same data is used in more than one place. (Morris, 2012, p. 99)
- **Audit and regulator experts** can speak behalf of the regulator who inspects the regulations and auditory requirements the organization is subjected to. This person may be also the business domain expert. (Morris, 2012, p. 101)

Integration middleware is used to integrate client organizations' business processes and their representatives can also be considered data stakeholders. Additional technical experts can inform the project of systems invisible to the data store owners or business domain experts. Bringing too many parties into the project introduces a risk of making the work unwieldy, but bringing in too few may result in overlooking some detail. It is up

to the migration project's judgement how wide of a net of stakeholders to include. (Morris, 2012, p. 102)

Managing key data stakeholders requires firstly recording them. Each legacy data store needs a data owner, a business domain expert and a technical data expert. For small data stores they all may be the same person. Each stakeholder is recorded in a centralized place with proper contact details. (Morris, 2012, p. 102)

Each data stakeholder should receive a personal briefing about the degree of the commitment expected. Data owners need be aware they are the ones signing off the decommissioning document. (Morris, 2012, p. 103)

4.4 System retirement plan

System retirement plan documents the requirements of the data migration that will allow the legacy data store to be decommissioned. The plan has to be tangible in a way the data owners and business domain experts can access. The plan is a list of organized requirements that must be fulfilled before the data can be migrated and the legacy data store turned off. The list should contain at least following items (Morris, 2012, pp. 127-139):

- **Training** on the target system is responsibility of the bigger programme the migration project is part of, but it is worth checking the business is getting the level of training it needs. Training may introduce some lag to the project timeline.
- **Testing** refers to the user acceptance tests the business performs before decommissioning can be signed.
- **Data retention** contains the rules telling which units of data will be migrated to the target system and which data classes can be left out. The less data to migrate the better. Data that is necessary to the business needs to be migrated somewhere, either to the target system or to an archiving software.
- **Data audit** is the verifiable proof that all the units of migration in the legacy data stores are accounted for in the migration.
- **Go-live restrictions** are the time windows suitable for the migration project to take place. Businesses often have limited periods of time when the legacy systems can be accessed and when all the stakeholders are available.
- **Customer experience** records how the migration affects the businesses actual customers and whether some customers need special attention.
- **Fallback** plan states the target system's essential parts it cannot function without. In case fallback to old system is required it should be made certain that the data has not been changed in the process.
- **Resources** are the people who are responsible for the sign-offs and support in the data quality rule process. Also relevant physical resources should be recorded.

- **Units of migration** must be agreed upon and the first-cut view of them will be created at the earliest interviews. For example, units of migration can be specified on a messaging pattern level as long as they are relevant to the business.
- **Transitional business processes** are the business processes occurring only because of the data migration project. They can be for example transactions that start in the legacy system but complete in the target.
- **Decommissioning certificate** needs to be front and foremost in the project teams mind. After the decommissioning certificate is signed the actual migration will take place.

Before the decommission certificate can be signed, all the headings in the system retirement plan needs to be signed off. That represents the data owners commitment that they have asked for everything they need the project to include. The retirement plan will change during the project, and its different headings will be signed off at different times. Data owners will sign it at least three times: firstly the initial plan, then after gap analysis and mapping, and finally after the target system is built and tested. The last sign off is the decommissioning statement after which the data migration can be completed and the legacy data store shut off. (*Morris, 2012, pp. 129-130*)

In an actively used middleware environment the legacy system can be used as a fallback method and the actual legacy system decommission will not take place immediately after the target system is running. There may be different go-live restrictions per customer which will affect the project decomposition.

4.5 Data profiling and landscape analysis

Landscape analysis is one of the first tasks in the project and can be used as a fast iteration before setting project budget and timeline. The goal of landscape analysis is to collect a list of all legacy data stores relevant to the migration project. The legacy data store list should contain at least the following items (*Morris, 2012, p. 125*):

- **Identity** is a name by which the data store is known in the organization. It should be a descriptive and understandable.
- **Key data stakeholders** name the persons in the roles of business domain expert and data owners. Technical system experts are also required where appropriate. Key data stakeholder management is explained in section 4.3.
- **Data quality** assessment for each legacy data store is recorded. It can be impressionistic or quantitative depending if a there has been a formal profiling process.
- **System retirement plan** records briefly the current situation the plan is in for the data store. System retirement plan is explained in section 4.4.

- **Technical details** on the level that is useful when getting data out of the store. Relevant things include the format, location and how to access it, quantity of data inside the store including number of bytes and data complexity. Data qualities like rate of change is relevant too. Record which conceptual entities the data represents and where they sit in the project.
- **Topography** information will include where each legacy data store gets its information from and where it sends it to. These relationships link the data stores together.

The documentation depends on the library services chosen for the project in migration strategy but a fully integrated set of services is preferred where documentation on the data stores is linked to the documentation on the data key stakeholders (Morris, 2012, p. 109). Larger data stores require greater analysis and the storage of middleware's configuration data is a key element in the landscape analysis. The documentation for these extensive data bases can be broken down in meaningful pieces, for example by unit of migration.

4.6 Managing and enforcing data quality rules

Data quality rules are a central project module, and majority of time, resource and effort will be spent there. Every data quality, selection or preparation issues should have a data quality rule and all data quality rules raised in any phase of the project should follow the process outlined here. A new rule can be raised by any stakeholder and creating them should be easy and accessible throughout the organization. The data quality rule document should contain at least the items listed in *Table 8*.

Table 8: Items in data quality rule document. (Morris, 2012, pp. 153-156)

Item	Content
Short name	Descriptive name easy to use in a conversation.
Cross reference	Reference to parent data quality rule.
Raised by	The person who initially discovered and raised the rule.
Legacy data store	Legacy data store's identifier as given in landscape analysis.
Date raised	Helps managing the process.
Identifier	Unique identifier allowing easy distinguishing from other items.
Priority	Suggested priorities are ' <i>must be fixed or project fails</i> ', ' <i>extremely useful to be fixed</i> ' and ' <i>cosmetic</i> '.
Status	Suggested statuses are ' <i>new</i> ', ' <i>open</i> ', ' <i>cancelled</i> ' and ' <i>completed</i> '.
Data quality assessment	Extended description of the problem and a <i>quantitative</i> assessment of the quality. The assessment of quality needs to be accompanied by a testable verification rule, a

	database query to check correct relations for example. Strengths are also recorded.
Method	Explains what actions the rule requires and what mitigation is going to take place to accommodate less than perfect data. If required the method section may consist of multiple tasks each specifying the person responsible and a deadline.
Metrics	Describes how the data quality rule progress will be measured, and when it will be considered complete.

When the rule is initially raised it is unlikely to contain all the information, and it is the data quality rule board's responsibility to prioritize, quantify and plan the rule. The data quality rule process is outlined in *Figure 5*.

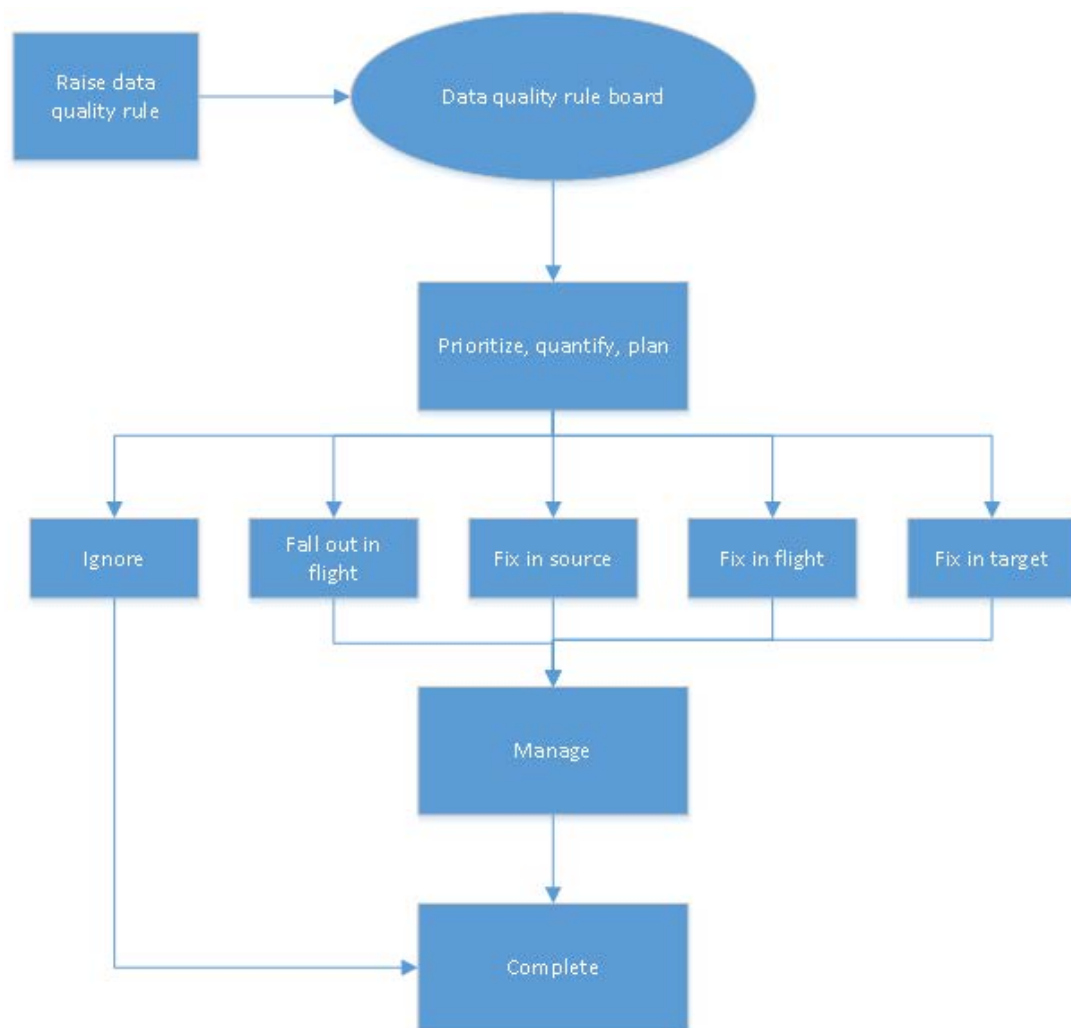


Figure 5: Data quality rule process. (Morris, 2012, p. 141)

There are multiple types of fixes introduced in the figure. *Fix in source* and *fix in target* differ also in timeline, as fixes in the target are done after data has been loaded on the target system. *Fall out in flight* is a case where writing the code for migrating very obscure and rare edge cases is more risky than doing the task manually, and the gaps are then ignored. *Fix in flight* is the most common solution besides *ignoring* the data gap. The data gap is fixed after extracting the data from legacy data store and before loading it to the target system. The data may be enriched from other data stores, transformed or consolidate before loading processes. (Morris, 2012, p. 143)

Data quality rule board consists of data migration analyst, business domain experts and technical data experts of each data legacy store. The responsibilities of the board include evaluating new rules, reporting on their progress, and reprioritizing and redirecting resources as necessary. The data quality rule process should be started as early as possible as rules can be generated already in the landscape analysis phase. (Morris, 2012, p. 151)

4.7 Gap analysis and mapping

Gap analysis and mapping module is where the tools and techniques for discovering and analysing data gaps are documented. Though there is a lot of software support available, frequently manual methods are used. Different types of data gaps are expanded below (Morris, 2012, pp. 158-160):

- **Reality gap** is a situation where data does not represent the real world. Uncovering these rely on the business domain experts' knowledge or audits and surveys. Audit or survey means collecting data from the business environment to reach an adequate level of quality.
- **Internal gap** means that data does not align with the data store rules. Some of these are discovered already in the landscape analysis module with data profiling tools.
- **Migration model gap** happens when data does not conform the legacy data rules even if it conforms the rules of its own data store.
- **Target model gap** refers to a difference between the migration model and the target model. Target model includes the rules of the target data store and system.
- **Topographical gap** comes to light when there is a missing link between data stores suggesting there is an additional data store.

Gap analysis module and mapping is closely related to data quality rules as any transformation and data quality monitoring is defined here. Data monitoring is relevant especially if fix in source type of fixes are used to prevent data from decaying during the project. (Morris, 2012, p. 163)

Data mapping brings together the requirements from system retirement plan, data quality rules, migration models, and landscape analysis. Data mapping links the fields in legacy data store to the ones in the target system using a common conceptual entity and leveraging the expertise of the team of data stakeholders. It is best to start linking conceptual entity to entity, then move down to a deeper level of granularity linking field to field. (Morris, 2012, pp. 163-165)

Mapping cannot always be done with one-to-one relationships and additional mapping rules need to be raised. Mapping rules are documented separately from data quality rules in a centralized location. Mapping rules may contain extraction specific rules determining the criteria by which the source data is selected. Navigation rules are required when data needs to be linked from multiple different tables or enriched from other sources. Validation rules are already defined in landscape analysis and raised as data quality rules, and they need to be run according to target model requirements. Transformation rules are almost always used after extract and prior to load. Loading rules show where the data will go in the target system. (Morris, 2012, pp. 163-165)

4.8 Migration end-to-end design

Migration end-to-end design is documented as a general approach describing the architecture from first extract to last legacy data store decommission. The content of the migration end-to-end design should include following items discussed when creating project strategy: *project scope, migration form, key business data areas and units of migration*. System retirement plan will provide the *audit and data lineage requirements, training requirements, testing, data retention, go-live restrictions, customer experience and units of migration*. (Morris, 2012, p. 177)

Additionally, following items should be documented in the end-to-end design (Morris, 2012, pp. 178-179):

- **Migration reporting** describes the level of reporting detail expected from extraction, transformation and load steps.
- **Technology overview** should provide an architectural diagram also showing responsibilities of different stakeholders where applicable. Different software described should include *data transportation, migration, transitional data store, master data management and data archiving software*.
- **High-level cutover timeline** should include the resource implications for the extraction, transformation, load and decommissioning. With a phased migration form a generic cutover timeline is needed.
- **Outline testing strategy** describes who will be performing the tests and when they will be performed.
- **Main sources and target data stores** defined in landscape analysis.

Extraction, transform and load design produces a tenable flow from legacy data stores to the target taking into account the analysis, requirements and non-functional considerations. The design contains functional and non-functional requirements. Non-functional requirements include *data sizing, runtimes, sequencing, hardware and network considerations* and *fallback*. (Morris, 2012, p. 181)

Fallback is the group of steps that will be taken to get an enterprise back into the position it was in prior to the data migration. A proper fallback plan is essential in the integration middleware environment where the go-live window may be short and client systems require continuous availability. Checkpoints are part of the fallback plan. A check point is a decision point where it is agreed a new system is stable enough to go forward with or from which fallback occurs. (Morris, 2012, p. 186)

The fallback can take different forms: partial fallback, full fallback, continue with caution or switch to a phased delivery. *Fall forward* is done on an occasion where falling back to legacy is not an option. When falling forward the minimum functionality is supported with the partially delivery system and with transactional business processes that will be later replaced. (Morris, 2012, pp. 187-188)

Functional requirements can be categorized under extract, transformation and load designs. Extract design describes how data is read from the legacy data stores using the legacy data store lists originating from landscape analysis and the mapping specifications including extraction rules and validation rules. Transformation design is the reformatting of data that occurs between extract and load. It will use the transformation rules generated in the data quality and mapping module. Load design specifies how the data is loaded into the target. (Morris, 2012, pp. 190-191)

4.9 Software selection

In the migration execution phase an extraction, transform and load software will be utilized. There are multiple different options for this software as well. Some are free and open source while some are solutions by well-known commercial vendors. (Butler Analytics, 2015)

More sophisticated products provide additional features and usually the feature set is so complex it is unlikely to be successfully replicated in a locally produced software (Morris, 2012, pp. 42-43). Below are some examples of possible software choices:

- **Microsoft Integration Services** is an enterprise-level data integration and data transformations solution. Its relevant features include cleaning, mining, and managing SQL Server objects and data. Integration Services support reading data from different sources such as XML data files, flat files, and relational data

sources, and loading the data into one or more destinations. (*Butler Analytics, 2015*)

- **Talend Studio** features connectors to natively connect multiple different databases, flat files and cloud-based applications. Talend provides a graphical drag-and-drop interface for designing, testing, and generating code. It is designed managing and monitoring simple, one-time extract, transform and load projects to complex, ongoing data synchronization projects. (*Butler Analytics, 2015*)
- **MetaSuite** is be used to extract, transform and merge operational data. It can be used to consolidate and rationalize data. A tool for data migrations, data conversions of replaced legacy applications and for data consolidation. (*Butler Analytics, 2015*)

Data profiling and quality tools are to be used in landscape analysis and data gap analysis modules. Optimally the data profiling and quality tools would be part of the migration controller so the any rules and maps generated in the previous phases can be utilized seamlessly during the migration execution. Obvious software selection criteria is compatibility with legacy and target data store formats.

5 DATA MIGRATION DESIGN FOR A MESSAGING PROCESS

5.1 Migration strategy

This migration plan presents a process for certain migration units selected from the legacy system. The goal is to provide a process for migrating a set of configuration data that make up an end-to-end messaging process. Migration execution is not performed in the scope of this thesis, and more defined process plan is created during the actual execution. Most of the plan is reusable for other migration phases sharing similar migration units.

Only configuration data will be migrated, transactional data and master data is be left out of scope. The initial scope's migration units consist of integration patterns that are represented as general conceptual models. On the outset the project is aware of one major legacy data source, an extensive relation database containing vast majority of the configuration data. More data stores will come up during landscape analysis.

The migration is done in *phases*. The project form will also be *parallel* and *always on*, so the legacy system will be running alongside with the target system and the legacy system cannot be turned off during the migration project. Project phasing will be based on customer organization and document type. The phases may be far apart from each other on the project timeline.

The data migration project will consist of in-house resources and no migration supplier will be used except for any migration controller software utilized. The division between technical and business resources is be done between different organizational units. Project consists of following modules that are listed in the order they are started:

- **Migration strategy and governance** – On the outset an initial migration strategy is built and governance processes are created. Target for the first phase is chosen. Required documenting libraries are instantiated and made accessible for project organization.
- **Data quality rule processes** – The data quality rule process is started and initial board members are chosen. Data quality rules may come up from every module, all of them are managed in this module.
- **Landscape analysis** – An amnesty about the migration project is published throughout the organization aiming to find the private data stores relevant to the migration phase and units of migration. Initial legacy data store is analysed for additional relationships and data stores. Each legacy data store needs a data owner, a business domain expert and a technical data expert. In some cases they may be the same person.

- **Key data stakeholder management** – The data stakeholder are recorded and briefed about their roles and expected commitment.
- **System retirement plan** – A set of requirements for disabling the legacy system process is gathered in co-operation with the data stakeholders.
- **Gap analysis and mapping** – Gap analysis may be performed when most legacy data stores are found and major data quality rules are set.
- **Migration design and execution** – The extract, transform and load processes are run after system retirement plan is signed off by business for the last time.
- **Go-live monitoring and fallback readiness** – Fallback plan is part of the migration design and will take place of the legacy decommission module.

Legacy decommissioning is not be part of the project even though described in Section 3.4. Instead, it is replaced by go-live monitoring and fallback readiness module. The fallback needs to be near instant in case of target system failure so the legacy system can take over the message processing.

Project is decomposed firstly by phase and secondly by migration unit. Initial lists including legacy data stores and stakeholders are defined after project decomposition is clear.

5.2 Conceptual entities

Figure 6 shows a message process overview on the legacy system. All processes follow the same pattern, so each component is executed in the same order in all processes. In the figure the process is split into two transactions at the beginning, each transaction taking the same processing steps. In legacy system's terms, one process contains one or more transactions.

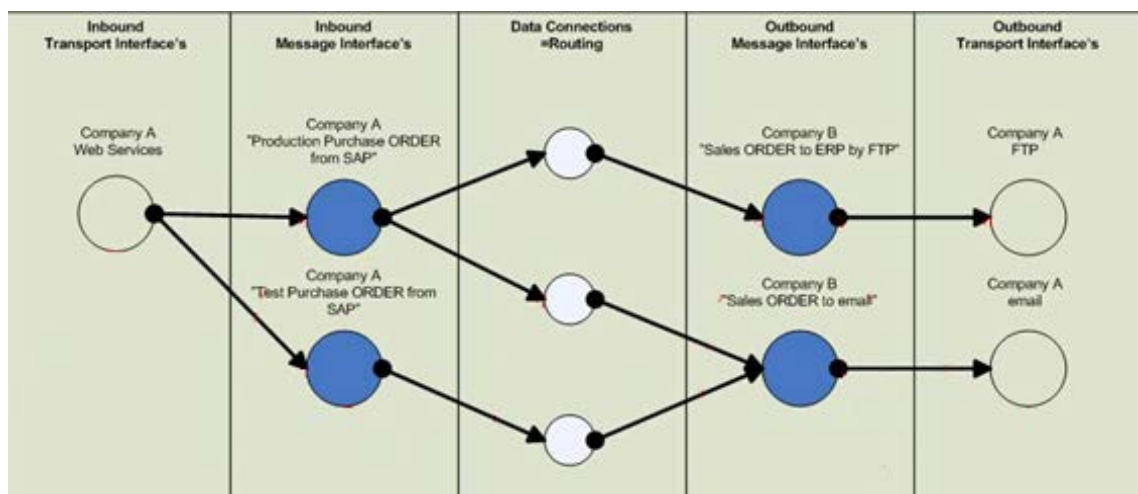


Figure 6: Source system conceptual entities in a messaging process. (Javanainen, 2009)

In the figure above there are five main concepts introduced and they each make up a conceptual entity used in the migration process. *Table 9* explains the different concepts, the design patterns they represent, and information relevant to them.

Table 9: Design patterns as conceptual entities in legacy system. (Javanainen, 2009)

Concept name	Pattern	Description	Information
Inbound Transport Interface (ITI)	Endpoint	ITI receives messages to service center.	Name, Owner company, ITI type, Routing method, Connected Inbound Message Interfaces, Invoicing settings, Support
Inbound Message Interface (IMI)	Translator	IMI receives messages from ITI.	Name, Owner company, Message type, Message format, Message format version, Message content type, System, Middleware, Processors, Parties, Routing rules, Message Connections
Message Connection (MC)	Router	Message Connection is a link between IMI and both OMI and OTI. Data connections are the routing rules inside the service center.	Name, Status, Activity, Project manager, Technical specialist, ITI and IMI, OMI and OTI, Advanced processing (delay / attachments), Transport frames, Processors, Invoicing settings
Outbound Message Interface (OMI)	Translator	Outbound message interface receives messages from the service center.	Name, Owner company, Message type, Message format, Message format version, Message content type, System, Middleware, Processors, Parties, Routing rules, Message Connections
Outbound Transport Interface (OTI)	Endpoint	OTI sends messages from the service center to an identified recipient.	Name, Owner company, OTI type, Routing method, Invoicing settings, Support

Target system is already well defined and the target model is used as the migration model. Target system consists of a group of micro services that collectively provide the complex integration features (*Liaison Technologies, 2014*). The target system provides concepts to achieve similar messaging patterns but the concepts also provide more flexibility. *Table 10* lists the most important concepts in the targets system, the patterns they represent, as well as the information relevant to them.

Table 10: Design patterns as conceptual entities in target system.

Concept name	Pattern	Description	Information
Gateway (GW)	Endpoint	Functions as the inbound connectivity end-point into the system.	Configuration
Task	Translator	Most granular unit of work in the integration flow. Is implemented as a script to perform a web service call, library method call or to decide the next pipeline to be executed etc.	Name, Service, Script, Post processing script
Activity (ACT)		Organizes tasks into an execution flow.	Enterprise, Level, Name
Pipeline (PL)	Channel	Defines the format and routing information of a message. Executes one or more activities that process the received message. Pipeline is endpoint specific.	Organization, Business process, Name, Document type, Document name, Document usage, Document delimiter set, Business identifiers
Connection (CON)	Router	Connects two or three pipelines: a mandatory set of an input pipeline and an output pipeline plus an optional connection pipeline that is executed between the mandatory pipelines.	Type, Name, Tags, Customer contract ID, Input PL, Connection PL, Output PL
Process (P)		Models the relationship and choreography of pipelines.	Enterprise, Organization, Name, Expectations script

These conceptual entities are connected in a way that supports the same integration features as the legacy model. The target entities will form the target model which is used as the migration model, so no additional models and mappings are required during the migration project.

5.3 Target model

The target system allows more flexible ways of creating message processing. Connecting pipelines in different ways and routing messages from tasks does not limit the system in

in to a certain sequence of actions for each message process. In the source system the process flow is the same for each process, all components will follow each other in the same order. For the purposes of migration the target model aims to be as close to one-to-one mapping with the legacy system as possible. The target model overview is illustrated in Figure 7. (Hell, Lehtonen, Mäntynen, & Pakkanen, 2016)

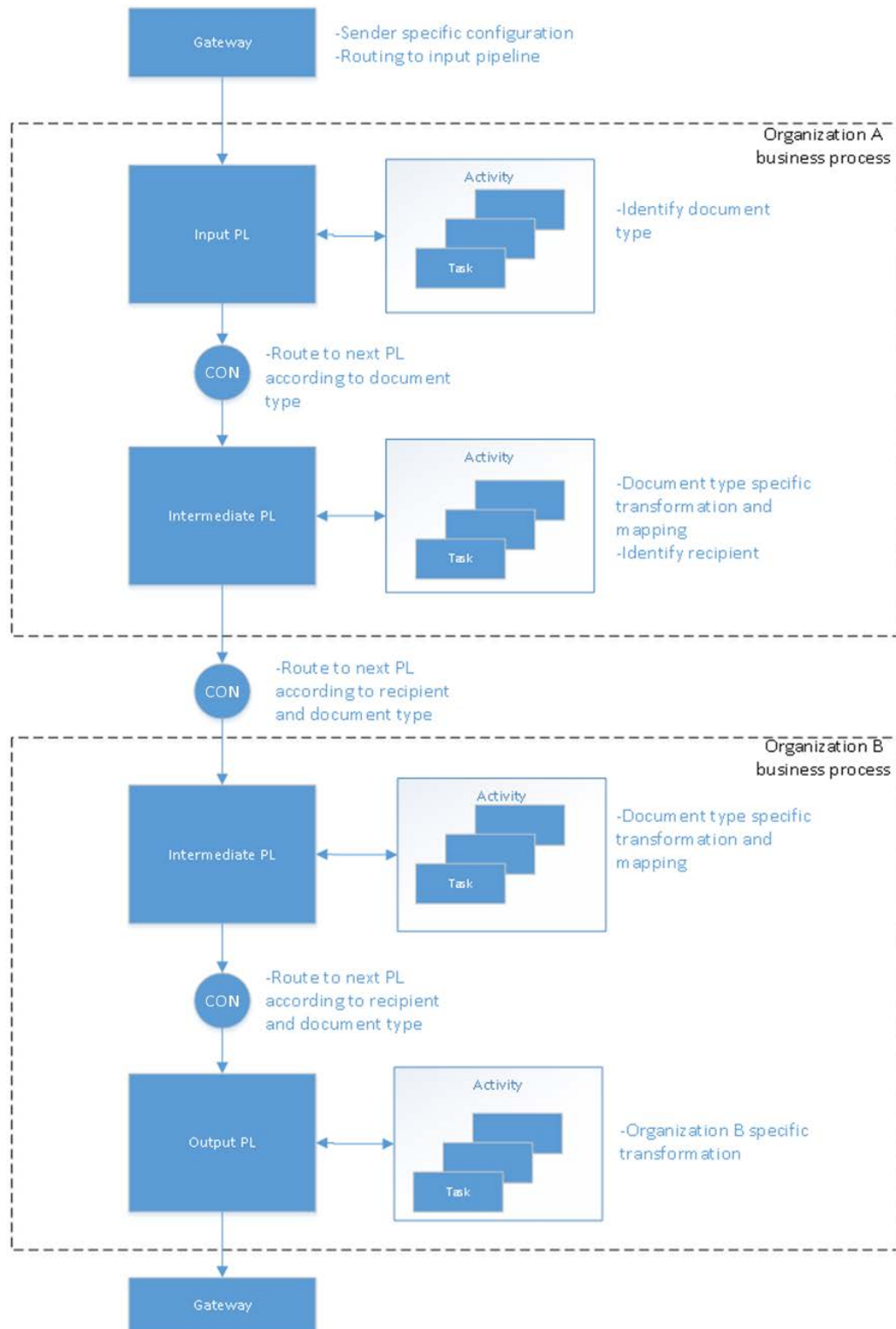


Figure 7: Target model overview for a migrated process.

In *Figure 7* there are four different pipelines: an input pipeline, two intermediate pipelines and an output pipeline. Each pipeline has an activity with a set of different tasks with their functionality explained in the figure. The pipelines are connected with connection concepts that are responsible for routing the message to the recipient according to the routing rules: document type and recipient identifier. (Hell, Lehtonen, Mäntynen, & Pakkanen, 2016)

There is a configuration related to each inbound gateway authenticated by user name and password pair. This configuration is used to route the inbound message to the input pipeline. (Hell, Lehtonen, Mäntynen, & Pakkanen, 2016)

The target model has fewer different concepts than the legacy system but they are assigned similar responsibilities, and migrated processes have a strictly defined order of execution. To proceed with gap analysis and mapping the conceptual entities need to be mapped.

5.4 Conceptual entity mapping

Data profiling is executed during landscape analysis. Gap analysis includes data quality and transformation rule creation as well as mapping rule generation. Data gaps are not fixed in source, so in-flight fixes will be favored to avoid any regression in the legacy system. *Table 11* below shows the conceptual entity mapping between source and target systems. The in-depth column-level mapping is done in the gap analysis and mapping module. From there the mapping and transformation rules will emerge.

Table 11: Concept mapping between the source and target system.

Legacy concept	Target concept	Conceptual gap
ITI	Input PL	
IMI	Sending organization's intermediate PL	
MC	CON	MC concept does not have an exact equivalent in the target model but the connection between the organizations business processes has similar responsibilities. Connections are used always when moving the processing from one pipeline to another whereas in the

		legacy system the fixed concepts are be linked directly to each other.
OMI	Receiving organization's intermediate PL	
OTI	Output PL	

Table 11 maps the legacy and target concepts and explains any gap there are on conceptual level. After entity level mapping is complete, a field to field mapping is done leveraging the data stakeholder expertise. This mapping process will create the transformations rules used in the migration end-to-end design.

5.5 Migration end-to-end design

Migration end-to-end design is part of the *migration design and execution* module. The design is an initial and general approach to the migration process, and low level extraction, transform and load designs are defined during the project.

The main legacy data source is a relational SQL database. The software responsible for extract functionalities can connect directly to the database with read-only permission. A phase and migration unit appropriate select statement is prepared.

Transitional database is not used. The units of migration are small and data sizing is expected to be relatively small and to be handled in-process. The migration units are chosen so that they can be separately loaded on to the target data store. Each conceptual entity is a single migration unit. *Figure 8* shows and overview of the extract, transform and load design.

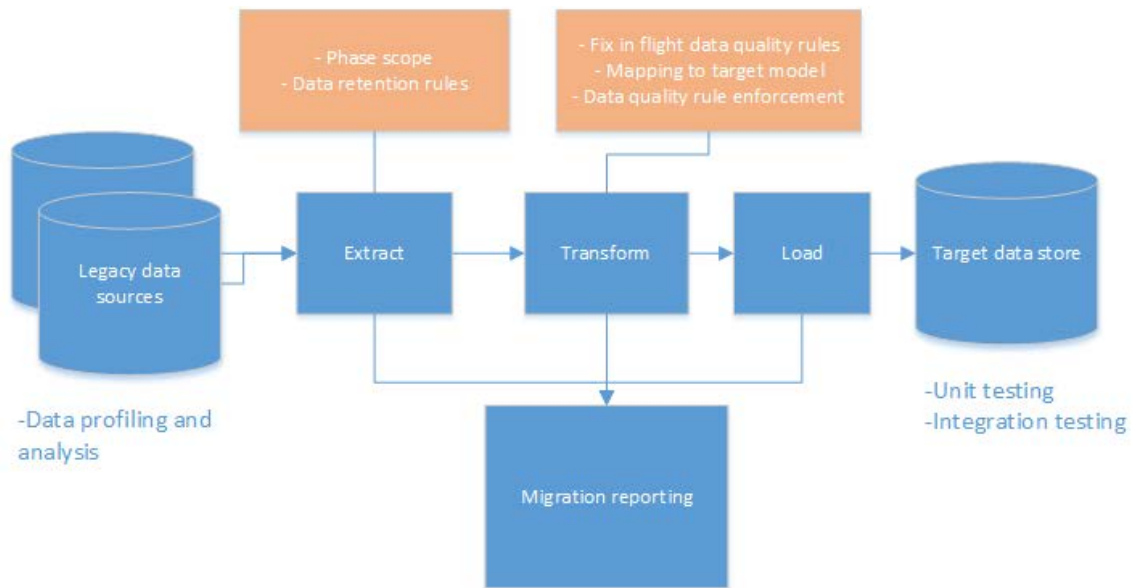


Figure 8: End-to-end design overview.

Testing single units of migration is possible in the target system when the phases are executed in an order that builds the messaging process incrementally. The cutover timeline needs to be defined when the project scope and scale is quantified.

The extract, transform and load tasks include following actions:

- **Extract** – Running the selection query for the phased migration units while taking data retention rules into account. Data is moved in manageable chunks from the source system into the migration controller without creating too much load on to the legacy data sources, as the legacy system is always on and runs in parallel with the target system. The reads should be done entity by entity and each entity needs to have its own selection query.
- **Transform** – Immediately after extracting the entities are transformed in to a format expected by the target system. Mapping rules and data quality rules are applied during the transformation process, and data may be enhanced and enriched. Cross-reference tables may be created beforehand manually when required.
- **Load** - *Configurator* service is used to load the data on to the target system. Configurator supports creating multiple objects at once and referencing previously created object data in the next objects to be created in the batch, so a unique identifier generated in the system can be referenced later in the execution.

Each step is accompanied by a fallback plan. The fallback plan is simple before any successful load operations: the transformation gap is fixed and the process is run again. A more sophisticated fallback plan is needed in case gaps are noticed after a successful

load, for example during the integration testing. Each step in the process needs to write an informative log file to provide a log file for migration reporting.

Figure 9 shows a simplified initial *extract, transform and load design* created in Talend Studio's graphical user interface. Leftmost there are three legacy data stores, one as the main data source and two as lookup sources to enrich the data or to create additional relations. These components represent the extract phase and each can have their own selection query and column filtering before passing the data on to the transformation step. There is likely to be a different set of data sources in the final iteration.

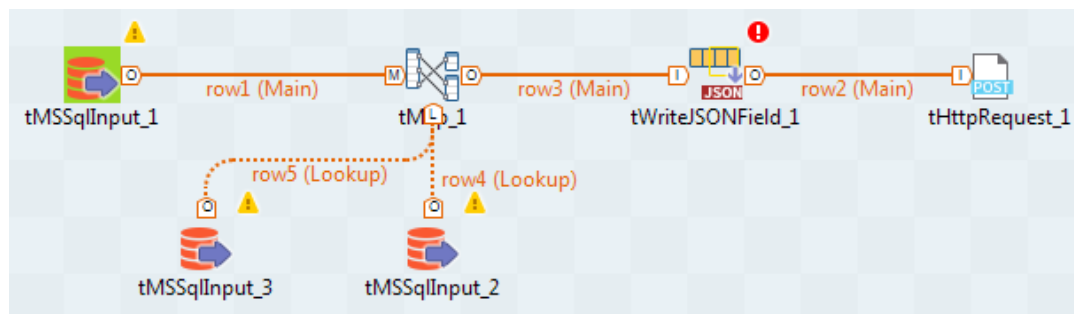


Figure 9: Graphical interface with simple extract, transform and load components in Talend Studio.

Connecting these three legacy data stores there is a mapping component that allows combining the input data into entities holding all the required data. Then using the mapping component the validation rules can be run against the entities that now should schematically follow the target model. A simple format transformation is done before using a web service to load the data on to the target system. The request is run with a debug parameter and the configurator will respond with a verbose message useful for migration reporting.

5.6 Go-live monitoring and fallback readiness

Go-live will happen in phases, and migration processes can be enhanced iteratively. Go-live is possible only after the final signoff by the business. During and after the cutover period there needs to be adequate monitoring to detect any message process failures. In case they are recurring and cannot be solved in a timely manner a fallback is needed. To allow quick fallback the legacy system will run in parallel with the target system for a set period of time. After the target system has run without major issues for a certain period it can be set on maintenance mode and fallback option will not be utilized anymore.

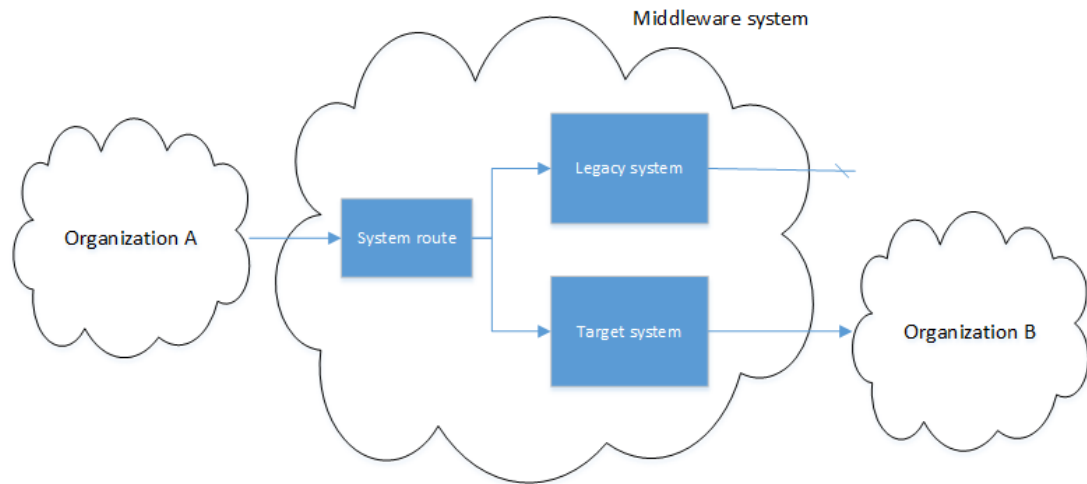


Figure 10: Fallback plan overview.

As seen in the *Figure 10*, messaging processing takes place in both systems simultaneously. In case the processing starts to fail in target system, the outbound connection may be turned on for the legacy system and turned off for the target system effectively changing the system the message goes through, but without any functional changes to the client systems. The messages that failed processing in the target system may be reprocessed in the legacy system and they will be delivered to recipient using the newly activated outbound transport interface.

In practise the *system route* component is the legacy system interface expanded to duplicate the incoming traffic also to the target system. The modification needs to be done in as non-intrusive way as possible. It is a part of the migration project strategy to decide on what extent the client organization will be involved.

6 EVALUATION OF THE MIGRATION DESIGN

6.1 Peer review

A peer review was performed as an open discussion about the project plan. The review board consisted of technology-oriented personnel working with legacy and next generation platform support and development tasks.

Data quality and mapping rule generation during the landscape analysis and gap analysis and mapping modules were perceived as the greatest challenge in the plan. Message processing requirements and specifications are likely outdated when documented and change request are scattered around different email chains and support tickets. A light-weight communication channel is required to keep the communication up the required level and to make sure migration project related documentation is kept up to date consistently. The business expert engagement is required to some extent, even if it is likely problematic to get these resources committed. Also, there are few technical experts in the organization familiar with both systems. *(Lehtonen, Thomsson, & Toivonen, 2016)*

The migration end-to-end design contains a rough plan to use a configurator service for loading the data on to the target system. Migrating a messaging process as a whole requires generating single file containing all configuration items and uploading it to the target system. Additionally, the items need to be ordered in a sequence that allows referencing backwards, so the next item to be created contains reference to the previously created item. This is a feasible solution for simpler messaging processes but complex integration cases may create unforeseeable issues in file size, mapping complexity, and referencing problems. *(Lehtonen, Thomsson, & Toivonen, 2016)*

The controller service provides a verbose response message containing information on which items were created and which ones had validation errors. A rollback configuration is provided in the response for all created items allowing a simple fallback plan where all the extraction, transform and load steps can be rerun. *(Lehtonen, Thomsson, & Toivonen, 2016)*

Unit testing will be problematic in the target system, especially if the message process is migrated in a one go. The target system does not provide testing for individual migrated entities and tests can be run only as integration tests. This may not necessarily prove to be a problem. If the message process is migrated in multiple steps, testing can take place after each migration step as long as the entities are migrated in the order they are executed in the final process. Even with this approach the testing procedures eventually build up into a full integration test. *(Lehtonen, Thomsson, & Toivonen, 2016)*

Overall, the migration project plan contains an extensive workload and many laborious manual tasks. New customers are on-boarded to the new system, and when major changes are required on the legacy system the processes may be rewritten on to the new system, and some migration processes can take place. Apart from this, the review board does not see the migration process as very beneficial without having any additional external driver. (Lehtonen, Thomsson, & Toivonen, 2016)

6.2 Architecture review

Target conceptual model is one of multiple ways to build a messaging process in the target system. An architectural review by target system development managers confirmed the approach is appropriate for the use case. Modifying the target model may allow simpler integration in the target system compared to legacy system in some cases. (Hölttä & Javainen, *Data migration project architecture review*, 2016)

The role of quality assurance is not very extensive in this project plan, and this raised concerns in the review. It needs to be possible to test the migration implementation including all quality rules in a pre-production environment and verify the outcome before executing the migration. The project plan does not differentiate between environments and it is left to be defined how to use testing, user-acceptance and production environments in the migration project. (Hölttä & Javainen, *Data migration project architecture review*, 2016)

As master data is left out of scope, the processes defined in the thesis created some concerns. Configuration data depends on master data by referring to it (e.g. configuration piece “XYZ” is owned by organization “ABC”). If it is decided to do a trial migration based on this thesis, it can be hard to verify the target system as the master data entities referred to by the configuration data are not available in the target system. (Hölttä & Javainen, *Data migration project architecture review*, 2016)

Engaging the most fitting stakeholders to the project will be challenging. It is likely to be problematic to get internal resources for two reasons: they are tied to revenue generating efforts and the migration project does not necessarily generate any revenue. Also, the nature of the migration project resource need is temporary and goes away after the migration is complete. (Hölttä & Javainen, *Data migration project architecture review*, 2016)

6.3 Future work

Migration design evaluation raised concerns that the level of commitment from business resources cannot be as high as required by the tools introduced in this thesis. The project team needs to design a fast and lightweight communication channel and process, which allows efficient exchange of information with the data stakeholders. If this is not feasible, some alternatives for project organization need to be found. Lack of stakeholder commitment will mostly affect *data quality rule* process, because the data quality rule board either does not have the required knowledge to make decisions.

After process details have been agreed upon, technology selection needs to take place. Technical details were not discussed in depth in this thesis, so additional planning is needed. There are multiple supporting software choices available that need to be evaluated in the technical context of the migration project. Also, documenting libraries need to be set up in the starting phase taking the requirements from stakeholder communication in to account.

Some form of a trial migration is likely to be beneficial to the project, and it will uncover unforeseen issues in the processes and technology choices. The processes can be refined and technical issues fixed before proceeding with the actual migration project. The commitment from business resources needs to be taken in to account already in the trial migration, and processes should be streamlined to allow fast iterations with minimal involvement of the business data stakeholders.

During the trial migration project technical resources become more experienced with the tools and techniques, likely allowing faster migration execution during the actual migration project. Also, data stakeholders can be informed about their roles more accurately. Trial migration helps setting the project deadlines and budget, and refining the initial scope, if needed.

7 CONCLUSIONS

The migration project plan presented is a high-level outline and becomes more defined after the project is started and the first modules and processes have begun. The project methodology applies to different types of migration projects but is discussed in the scope of configuration data on messaging middleware systems.

A significant portion of migration projects exceed their budgets or entirely fail. A proper project plan and strategy prevents a majority of the issues in the outset. This thesis uncovers the extensiveness of the tasks related to a successful migration project, including the level of engagement required by the business side of the project. All non-technical decisions are made by the business, and the technical side is mainly responsible for moving and transforming the data as well as assuring the quality and providing fallback methods in case of failure. Most of the processes take place before the actual data extraction, transformation and loading steps. Most important tools and techniques are provided to uncover the legacy data stores and creating the data quality, transformation and mapping rules.

The metadata model technique is discussed in more depth and initial conceptual entities and models are created. These models are mapped on a high level and are there to support field to field mapping that is done during the actual migration project. This is very likely to be a useful model for the actual project execution for the future.

Technical details need to be better defined. Reliable testing, monitoring and fallback processes are likely to be the most challenging features to be implemented to the migration controller and to the systems included in the migration project. Next steps to defining the migration project include a trial migration that engages more of the organizations business side and familiarizes the programme with the tools and techniques. The trial migration is also likely to refine and change the techniques and processes presented in this thesis. After the trial migration it is easier to set the project scope and budget.

As mentioned, there are issues left out of the scope of this thesis that need to be discussed during the migration project, and some of these issues came up in the project evaluation. Master data management needs to be addressed when creating the project strategy. Also, the project methodology concentrates on processes dealing with data-related issues and less attention is given on project management and governance.

Project evaluation also raises an issue about the availability of in-house resources, because the project is likely to not generate any revenue. Also, technical resources with

proper knowledge and experience with both systems are scarce. The project requires commitment from the technical resources and mapping tasks are extensive. Peer review raised some discontent with the migration project due to the laborious tasks especially taking place in the gap analysis and data mapping modules.

So, engaging the required resources and getting all stakeholders committed to the assigned tasks is likely to present challenges that are not adequately discussed in this thesis alone. After all, data migration is a business not a technical problem.

REFERENCES

- Butler Analytics*. (2015, 6 8). Retrieved from <http://www.butleranalytics.com/30-etl-tools/>
- Calton Pu, A. S.-K. (2011). An Observation-Based Approach to Performance Characterization of Distributed n-Tier Applications. *Future Generation Computer Systems*, (pp. 871-879).
- Chung, L., & Leite, J. C. (2009). On Non-Functional Requirements in Software Engineering. *A.T. Borgida et al. (Eds.): Mylopoulos Festschrift* (pp. 363-379). Berlin: Springer.
- Clement Dephine, G. S. (2010). Data quality as a key success factor for migration projects. *ICIQ*.
- Connelly, D. (1999). *Best Practices in Business Software Integration and the Open Applications Group*. Wirtschaftsinformatik 41.
- Erasala, N. Y. (2003). Enterprise Application Integration in the electronic commerce world. *Computer Standards & Interfaces* 25 (pp. 69-82). Ohio: Elsevier.
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison Wesley.
- Hell, J., Lehtonen, M., Mäntynen, T., & Pakkanen, J. (2016, 4 6). Alloy lab workshop. (P. Lehtinen, Interviewer)
- Hohpe, G., & Bobby, W. (2006). *Enterprise integration patterns*. Addison-Wesley.
- Hosseing, M., & Ardeshir, B. (2013). Toward a Comprehensive Framework for Evaluating the Core Integration Features of Enterprise Integration Middleware Technologies. *Journal of Systems Integration, Vol 4, No 1*, 1-29.
- Howard, P. (2011). *Data Migration*. A White Paper by Bloor Research. Retrieved from <http://www.bloorresearch.com/research/white-paper/data-migration-2011-p1/>
- Hölttä, L., & Javanainen, M. (2016, 5 9). Data migration project architecture review. (P. Lehtinen, Interviewer)
- Hölttä, L., & Javanainen, M. (2016, January 19). Legacy and next generation middleware architecture and design. (P. Lehtinen, Interviewer)
- Javanainen, M. (2009). *MANAGING TRANSACTIONAL INTEGRITY IN MESSAGING MIDDLEWARE SOFTWARE*. Tampere: TUT.
- Kalakota, R. R. (1999). *E-Business: Roadmap for Success*. Massachusetts: Addison Wesley.
- Lehtonen, M., Thomsson, R., & Toivonen, J. (2016, 05 11). Data migration project plan peer review session. (P. Lehtinen, Interviewer)
- Liaison Technologies. (2014). *Liaison Cloud Integration Platform: The Next Generation* (Confidential).
- Medjahed, B., Benatallah, B., Athman Bouguettaya, Ngu, A. H., Elmagarmid, A. K., & Medjahed, B. (2003). Business-to-business interactions: issues and enabling technologies. *The VLDB Journal* (pp. 59-85). Springer-Verlag.

- Morris, J. (2012). *Practical Data Migration*. bcs.
- Puschmann, T., & Alt, R. (2001). Enterprise Application Integration - The Case of the Robert Bosch Group. *Proceedings of the 34th Hawaii International Conference on System Sciences*, 1.
- Qureshi, K. A. (2005). *Enterprises Application Integration*. Islamabad: IEEE ---- 2005 Internatiol Conference on Emerging Technologies.
- Scott, C. R. (2004). *Project Management for Data Conversions and Data Migrations: A Data Conversion Methodology and Guide to Converting Data for Mission Critical Applications*. Bloomington, Indiana: Authorhouse.
- Tsur, S. (2001). Are Web Services the Next Revolution in E-Commerce. *Proceedings of the 27th VLDB Conference*. Roma, Italy.
- Xu, L. D. (2011). Enterprise Systems: State-of-the-Art and Future Trends. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, VOL. 7, NO. 4, (pp. 630-640).